Objective

1.(B) Arithmetic Exception

2.(B) start exception finally end

3.(A) car maruti tata

4. (A) MyException

5.(A)

6.(B) runtime error

7.(c) slidebar

8.(A)

9.(C)

10. (C)

Subjective

1.

S.N	ArrayList	Linked List
1	ArrayList contains array internally	LinkedList contains queue
2	Manipulation is slow	Manipulation is fast
3	It works better when data is stored	It works better when the manipulation of stored data is required.
4	ArrayList implements list only	Linked List implements List as well as Queue.

5	Searching is fast	Searching is slow

2. JSP error page is used to handle the exception which occurs at run time . This occurs anytime in your web application.
   There are 2 methods to perform exception handling in web application
   (i)errorPage attributes of page directive.
   (ii) by <error-page> element in web.xml file.

3. Tags available in jsp
   1) <% java code %> Scriptlet tag
   2) <%@ dir-type dir-attr %> Directive tag
   3) <form> form tag
   4) <meta >

4. HTML is stateless because everytime client retrieves a webpage , the client opens a separate connection to the web server and the server does not keep any record of previous client requests.
Example: hit counter

5. There are many ways to iterate over list
   1) For Loop
   2) Enhanced For loop
   3) iterator
   4) While loop

FailFast aborts the operation as soon as it exposes failure and stops the entire operation.
Use Fail Safe to overcome the issue. Fail Safe doesn't abort the operation in case of failure.

6. HttpSession getSession() : returns the current session associated with this request , else create one

getSession(true) will check whether a session already exists for the user. If yes returns the object else create the new session

getSession(false)will check the existence of the session . If exists then returns the reference else returns null.

7. When we create a memory in a heap and forget to delete it then the memory leakage occurs .
Prevention: Use referenced objects to avoid memory leaks .
Code Reviews.
Check warnings

8. Yes final class is class that can’t be extended .

Coding:

```
import java.util.Scanner;

public class Ajeet_Shakya {

    static int MAX = 256;

    static boolean canBeFormStr2(String str1, String str2)
    {

        int[] count = new int[MAX];
        char []str3 = str1.toCharArray();
        for (int i = 0; i < str3.length; i++)
            count[str3[i]]++;
```

```
            char []str4 = str2.toCharArray();
            for (int i = 0; i < str4.length; i++) {
                if (count[str4[i]] == 0)
                    return false;
                count[str4[i]]--;
            }
            return true;
        }

        static public void main(String []args)
        {
            Scanner sc=new Scanner(System.in);
            System.out.println("Enter String 1");
            String str1=sc.nextLine();
            System.out.println("Enter String 2");
            String str2=sc.nextLine();
            if (canBeFormStr2(str1, str2))
                System.out.println("true (we can form string 2 using data
from string 1)");
            else
                System.out.println("false (we cannot form string 2 using
data from string 1)");
        }
}
```

2.

```
import java.util.*;

public class Ajeet_Shakya {
    public static void main(String args[]) {
     Scanner sc=new Scanner(System.in);
     System.out.print("Enter the number of elements you want to store: ");
     int a=sc.nextInt();

     int [] arr=new int[10];
     for(int i=0;i<=a;i++)
     {
          arr[i]=sc.nextInt();
     }

     System.out.println("Enter Sum");
     int sum=sc.nextInt();
     combination(arr,sum);
    }

    public static void combination(int[] arr, int sum)
    {

        int count = 0;
         Set<Integer> a = new LinkedHashSet<Integer>();

        for(int k =0;k<arr.length ; k++){
            for(int i =0;i<arr.length;i++)
            {
                if(sum == arr[k] + arr[i])
                {
                    a.add(arr[k]);
                    a.add(arr[i]);
                    System.out.println(a);
                    count ++;
                    a.clear();
```

```
            }
          }
        }
        if(count == 0 ){
          System.out.println("can not get values for given sum") ;
        }
    }
}
```